Archive of SID

ارائه یک استراتژی زمان‌بندی وظایف به‌منظور بهبود خصوصیات کیفی در محیط محاسبات ابری

مینا یزدانبخش پوده[۱]، کارشناسی ارشد؛ ریحانه خورسند مطلق اصفهانی[۲]، استادیار

۱- دانشگاه آزاد اسلامی واحد دولت آباد - گروه کامپیوتر - اصفهان – ایران، st_yazdanbakhsh@yahoo.com

۲- دانشگاه آزاد اسلامی واحد دولت آباد - گروه کامپیوتر - اصفهان – ایران، R.khorsand@iauda.ac.ir

چکیده: در طول سال‌های اخیر، یکی از جنبه‌های مهم محاسبات ابری زمان‌بندی پویای تعداد زیادی درخواست‌های وظیفه است که با نرخ متغیر به‌وسیله کاربران ارسال می‌شوند. زمان‌بندی وظیفه یک نقش کلیدی در سیستم‌های محاسباتی ابر بازی می‌کند و این نوع زمان‌بندی بر اساس یک معیار تنها نمی‌تواند انجام شود بلکه تعداد زیادی قوانین و شرایط به‌صورت یک توافق بین کاربران و فراهم‌کنندگان ابر باید در نظر گرفته شوند. درواقع این توافق، کیفیت سرویسی است که کاربران از فراهم‌کنندگان انتظار دارند. مراکز داده ابر نه‌تنها باید این وظیفه‌های عظیم را اجرا کنند بلکه باید نیازمندی‌های چندگانه کاربران مختلف را ارضاء کنند. در این مقاله، یک استراتژی زمان‌بندی وظایف چندهدفه با استفاده از مرتب‌سازی نامغلوب، محاسبه نرخ نرمال و آستانه ارائه می‌شود. هدف از روش پیشنهاد شده در نظرگیری تعدادی از مهم‌ترین معیارهای کیفیت سرویس در زمان اجرای وظیفه‌ها یعنی مهلت زمانی و هزینه می‌باشد. به‌علاوه، خصوصیت کشسانی ابر در نظر گرفته می‌شود. نتایج شبیه‌سازی بهبود را در شرایطی روی زمان تکمیل کلی، هزینه، میانگین بهره‌وری ماشین مجازی و نقض مهلت زمانی در مقایسه با الگوریتم‌های MultiObjective، FCFS، Min-Min، Priority Scheduling و MOF نشان می‌دهد.

واژه‌های کلیدی: محاسبات ابری، زمان‌بندی چندهدفه، خاصیت کشسانی، کیفیت سرویس.

A Task Scheduling Strategy to Improve Qualitative Features in the Cloud Computing Environment

M. Yazdanbakhsh[1], M.Sc; R. Khorsand[2], Assistant Professor

1- Department of Computer Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran, Email: St_yazdanbakhsh@yahoo.com

2- Department of Computer Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran, Email: R.khorsand@iauda.ac.ir

Abstract: Over the recent years, one of the important aspects of cloud computing is the dynamic scheduling of a large number of task requests which are submitted with variable rate by users. Task scheduling plays a key role in cloud computing systems, and this type of scheduling can not be done on a single criterion, but many rules and conditions must be considered as an agreement between users and cloud providers. In fact, this agreement is the quality of the services that users expect from providers. Cloud data centers should not only execute these huge tasks, but also should meet the multiple needs of different users. In this paper, a multi-objective task scheduling strategy is proposed using non-dominated sorting, calculate normal and threshold rates. The aim of the proposed approach is considering some of the most important criteria for quality of service at the time of tasks execution, that means deadline and cost. In addition, the cloud elasticity property is considered. The simulation results show improvement in the conditions of makespan, cost, mean utilization of virtual machines and deadlines violation compared to MultiObjective, FCFS, Min-Min, Priority Schedulig and MOF approaches.

Keywords: Cloud computing, Multi-objective scheduling, Elasticity, Quality of service.

تاریخ ارسال مقاله: ۱۳۹۶/۹/۶

تاریخ اصلاح مقاله: ۱۳۹۶/۱۲/۲۱

تاریخ پذیرش مقاله: ۱۳۹۷/۳/۱۹

نام نویسنده مسئول: ریحانه خورسند

نشانی نویسنده مسئول: ایران - اصفهان – دانشگاه آزاد اسلامی واحد دولت آباد.

www.SID.ir

Archive of SID

۱ -مقدمه

محاسبات ابری یک تکنولوژی روبه‌پیشرفت است که به‌طور کامل به اینترنت تکیه می‌کند [۱]. در آن‌همه داده‌ها و برنامه‌های کاربردی روی یک مرکز داده مستقر هستند. این مرکز داده شامل یک استخر بزرگی از سرویس‌دهنده‌های توزیع شده است که برنامه‌های کاربردی مختلف را میزبانی می‌کنند و همچنین تسهیلات پردازشی و ذخیره‌سازی متفاوتی را برای مشتریان فراهم می‌کنند. فراهم‌کنندگان ابر مدل پرداخت به میزان استفاده را برای استفاده منابع خودشان تطبیق می‌دهند [۲]. از سوی دیگر روی اینترنت، مشتریان به‌منظور استفاده از سرویس‌های مختلف ابری نیاز دارند به مرکز داده متصل شوند و بر طبق تقاضا سرویس‌های خود را تحویل بگیرند.

زمان‌بندی وظیفه یک روش پایه‌ای در محاسبات ابری است که توزیع وظایف محاسباتی در میان استخر منابع مجازی را به عهده دارد و یک موضوع مهمی است که تاکنون در آن زمینه تحقیقات زیادی انجام شده است [۳]. درحالی‌که فراهم‌کننده سرویس ابر و سرویس‌گیرنده‌ها اهداف و نیازمندی‌های متفاوتی دارند ولی هدف بیشتر روش‌های زمان‌بندی که تاکنون ارائه شده، بهینه‌سازی یک هدف می‌باشد [۴، ۵]. به‌علاوه نیازمندی‌های چندگانه ممکن است ماهیت متضاد داشته باشند. برای نمونه، حداقل کردن زمان تکمیل کلی و نقض مهلت زمانی روی تعداد زیادی وظایف درحالی‌که هزینه‌ها کاهش پیدا کند، دشوار است. در کل، مسئله‌های زمان‌بندی چندهدفه می‌توانند به دو گروه تقسیم شوند: روش‌های مبتنی بر اولویت با وزن تخصیص داده شده به هر هدف (روش‌های کلاسیک) و روش‌های شامل یک مجموعه از راه‌حل‌های نامغلوب. در یک مسئله بهینه‌سازی چندهدفه هر راه‌حلی بر طبق بیش از یک تابع هدف اندازه‌گیری می‌شود و هرکدام باید حداقل یا حداکثر باشند. روش‌های کلاسیک در مقایسه با روش‌هایی که از راه‌حل نامغلوب استفاده می‌کنند دارای ضعف‌های اساسی می‌باشند. به‌طوری‌که ۱) این روش‌ها قادر به جستجوی تمام فضای مجاز مربوط به مسئله نیستند. ۲) این روش‌ها به‌تنهایی یک روش هوشمند به‌حساب نمی‌آیند و ۳) توابع هدف مربوط به این روش‌ها برای جمع شدن باهم نیازمند نرمال شدن هستند.

یکی از مهم‌ترین چالش‌های موجود در محیط ابر، مقیاس‌پذیری منابع ابری است که اجازه تأمین بهنگام منابع را برای ملاقات نیازمندی‌های برنامه کاربردی می‌دهد. ازآنجایی‌که در زمان‌بندی الگوهای حجم کاری منظم نیستند بنابراین عملیات مقیاس‌بندی یعنی افزایش یا کاهش مقیاس منابع باید به‌صورت بلادرنگ و با حداقل دخالت انسان انجام گیرد تا منابع در اسرع وقت برای اجرای وظایف تأمین شوند. به این نوع مقیاس‌بندی خودکار خاصیت کشسانی[1] در ابر گفته می‌شود.

بیشتر روش‌هایی که تاکنون برای زمان‌بندی وظایف در بین ماشین‌های مجازی مختلف مراکز داده ابر ارائه شده‌اند، خاصیت کشسانی ابر را به‌صورت پویا در نظر نمی‌گیرند[۶–۸]. برای حل مشکلات مطرح شده در این مقاله یک استراتژی زمان‌بندی وظایف برای بهبود خصوصیات کیفی ارائه شده است که از خصوصیت کشسانی منابع ابر استفاده می‌کند.

در ابتدا با ورود درخواست‌ها با نرخ متغیر، تعداد ماشین‌های مجازی براساس بار وظایف رسیده مشخص می‌شوند. سپس ماشین‌های مجازی با در نظر گرفتن نیازمندی‌های کاربر و وظایف به‌صورت چندهدفه مرتب خواهند شد و وظایف با در نظر گرفتن ظرفیت ماشین‌های مجازی، به آن‌ها به‌طور پویا اختصاص داده شده و اجرا می‌شوند. لازم به ذکر است در حین اجرا، با توجه به بار وظایف رسیده و با استفاده از خصوصیت کشسانی ابر در صورت نیاز ماشین‌های مجازی جدید اجاره و در صورت عدم نیاز، آزادسازی می‌شوند.

در ادامه بخش ۲ یک مرور کلی از کارهای مرتبط فراهم می‌کند. بخش ۳ روش پیشنهادی که به‌صورت چندهدفه و پویا عمل می‌کند را ارائه کرده است. بخش ۴ ارزیابی عملکرد استراتژی پیشنهادی را فراهم می‌کند. نتیجه‌گیری و کار آینده در بخش ۵ بحث شده است.

۲ -کارهای مرتبط

در محیط ابر، زمان‌بندی وظیفه برای انتخاب و تخصیص وظایف به منابع مناسب برای ارتقای کیفیت سرویس استفاده می‌شود. زمان‌بندی وظیفه می‌تواند به‌صورت دستی توسط یک کاربر یا به‌طور خودکار انجام شود [۹]. در زمان‌بندی وظیفه دستی قبل از راه‌اندازی یک وظیفه، هر وظیفه به یک موتور اجرایی اختصاص داده می‌شود. نوع دیگری از عملیات زمان‌بندی، زمان‌بندی‌های خودکار است که به‌صورت پویا منابع موجود را به وظایف در زمان اجرا اختصاص می‌دهد. از سوی دیگر الگوریتم‌های زمان‌بندی را می‌توان مبتنی بر پارامترهای بهبود در محیط محاسبات ابری طبقه‌بندی کرد. از قبیل الگوریتم‌های زمان‌بندی مبتنی بر توازن بار، هزینه، اولویت، کاهش زمان تکمیل کلی، مصرف انرژی، سازگاری و غیره.

بسیاری از محققان، الگوریتم‌هایی در راستای بهبود توازن بار در محیط محاسبات ابری در نظر گرفته‌اند. تانگ و همکاران [۴] الگوریتم توازن بار جدید بر اساس هوش ازدحامی ارائه کرده‌اند که مسئله توازن بار را بر اساس تعداد وظایف و ظرفیت ماشین‌های مجازی انجام می‌دهد و درنهایت می‌تواند بار سیستم را متوازن کند و نیازمندی‌های مشتریان را برآورده کند. از اشکالات این الگوریتم استفاده از زمان‌بندی تک هدفه می‌باشد که یکی از مسائل مورد بررسی در این تحقیق است. در همین راستا بابو و ساموئل [۱۰] یک اصلاح در الگوریتم کلونی زنبورعسل برای توازن بار کارآمد و مؤثر در محیط محاسبات ابری پیشنهاد کردند که در آن رفتار جستجوگر زنبورعسل برای توازن بار در سراسر ماشین‌های مجازی استفاده می‌شود. هند گامال و همکاران [۱۱] الگوریتم زمان‌بندی گروهی (GTS) را پیشنهاد کردند که برای زمان‌بندی وظایف در شبکه محاسبات ابری با استفاده از کیفیت سرویس برای رفع نیازهای کاربر مورد استفاده قرار می‌گیرد. این الگوریتم با استفاده از معیارهای عملکرد متفاوتی از قبیل زمان اجرا،

www.SID.ir

Archive of SID

تعادل بار، متوسط تأخیر برای اولویت فوری مورد انتظار و متوسط تأخیر برای وظایف طولانی ارزیابی می‌شود.

در جهت بهبود هزینه در محیط محاسبات ابری لاکرا و کومار یاداو [۱۲] یک الگوریتم زمان‌بندی وظیفه چندهدفه برای نگاشت وظایف به ماشین‌های مجازی پیشنهاد کرده‌اند که باعث بهبود توان مرکز داده و کاهش هزینه بدون نقض توافق‌نامه سطح سرویس برای یک نرم‌افزار در محیط SaaS ابر می‌شود. از عملیات لازم در راستای این الگوریتم در نظر گرفتن اجاره و آزادسازی ماشین مجازی جدید می‌باشد که در تحقیق ما مورد بررسی قرار گرفته است. یک مکانیزم مقیاس‌بندی با اجراکننده فوق‌العاده حرفه‌ای (Suprex) توسط اصلان‌پور و همکاران [۱۳] با رویکرد هزینه ارائه شد. نتایج شبیه‌سازی نشان می‌دهد که Suprex می‌تواند کاهش هزینه‌های اجاره منابع برای ارائه‌دهندگان نرم‌افزار را کاهش دهد درحالی‌که زمان پاسخ و کاهش نقض مهلت زمانی را بهبود می‌بخشد.

در خصوص الگوریتم‌هایی برای کاهش زمان تکمیل کلی با هدف دستیابی به توازن بار در سراسر ماشین‌های مجازی و برای به حداکثر رساندن توان عملیاتی، بابو و کریشنا [۱۴] یک الگوریتمِ توازن بار الهام گرفته از رفتار زنبورعسل (HBB-LB)[۲] را ارائه کردند. روش پیشنهادی آن‌ها اولویت‌های وظایف بر روی ماشین مجازی را متوازن می‌کند به‌گونه‌ای که مقدار زمان انتظار وظایف در صف حداقل می‌شود. بر اساس مطالعه جامع از تأثیر الگوریتم زمان‌بندی وظیفه Max-Min بهبود یافته در محاسبات ابری، بهویی و رامانوج [۱۵] در این الگوریتم زمان اجرای مورد انتظار را به‌جای زمان تکمیل به‌عنوان پایه انتخاب می‌کنند.

در خصوص الگوریتم‌هایی که از خاصیت کشسانی استفاده کرده‌اند، بلتران [۱۶] در مقاله خود یک روش کشسانی جدید را تعریف کرده است که قادر است چهار جزء اصلی شامل مقیاس‌پذیری، دقت، زمان و هزینه را مستقل از سطح سرویس (زیرساخت، سکو[۳] یا نرم‌افزار) در نظر بگیرد. علاوه بر این، نویسندگان یک روش تجزیه و تحلیل برای ارزیابی رفتار کشسانی سرویس و یک ابزار سنجش برای خودکار کردن تحلیل ارائه کردند. گارگ و همکاران [۱۷] یک چارچوب و یک سازوکار پیشنهاد کردند که کیفیت و اولویت‌بندی سرویس‌های ابر را اندازه‌گیری می‌کند. چنین چارچوبی می‌تواند تأثیر قابل‌توجهی داشته و رقابت سالم بین سرویس‌دهندگان ابر برای برآورده کردن توافقنامه سطح سرویس (SLA) و بهبود کیفیت سرویس ایجاد کند. آن‌ها یک مکانیزم رتبه‌بندی مبتنی بر فرایند تحلیلی سلسله مراتبی (AHP)[۴] پیشنهاد کردند که می‌تواند سرویس‌های ابر را بر اساس الزامات کیفیت سرویس ارزیابی کند.

یک روش تأمین منبع پویا توسط قبائی و همکاران [۱۸] مبتنی بر مفهوم حلقه کنترلی MAPE (مانیتور[۵]- تجزیه و تحلیل[۶]- نقشه[۷]- اجرا[۸]) ارائه شده است. در روش پیشنهادی آن‌ها هدف مقابله با شرایط نامطلوب بیش از حد منابع و یا کمبود منابع است. این روش به‌طور منظم در فواصل زمانی مشخص اجرا می‌شود و مقدار مطلوب منابع مورد نیاز به هر بار کاری سرویس ابر را محاسبه می‌کند. نتایج تجربی نشان می‌دهد که روش آن‌ها هزینه کل و تعداد نقض توافقنامه‌های سطح سرویس را کاهش می‌دهد و بهره‌وری منابع را در مقایسه با سایر روش‌ها افزایش می‌دهد. همچنین این نویسندگان [۱۹] یک روش تأمین منابع ترکیبی برای خدمات ابر ارائه داده‌اند که بر مبنای ترکیبی از مفهوم محاسبات خودمختار[۹] و یادگیری تقویتی[۱۰] (RL) است. همچنین، یک چارچوب برای تأمین منابع مستقل ارائه می‌دهند که الگوی آن از مدل لایه‌ای ابر است. برای پیاده‌سازی روش پیشنهادی، آن‌ها یک چارچوب تأمین منابع ارائه می‌دهند که از حلقه کنترلی MAPE پشتیبانی می‌کند. روش آن‌ها به‌طور منظم در فواصل زمانی مشخص اجرا می‌شود و مقدار مطلوب منابع را با توجه به بار کاری سرویس‌های ابری تغییر می‌دهد.

تحقیقات قبلی برای زمان‌بندی وظیفه در محیط ابر دارای محدودیت‌های قابل‌توجهی هستند. درواقع بهینه‌سازی اهداف متداخل در زمان‌بندی وظایف به خاطر شرایط زمان اجرای نامشخص از قبیل خاصیت کشسانی روی منابع و توزیع بار کاری متغیر دشوار است. همچنین به دلیل اینکه ابر یک محیط پویا است، تغییرات در آن همیشه با توجه به شرایط در حال رخ دادن است. بنابراین استفاده از الگوریتم‌های زمان‌بندی که به‌صورت پویا عمل می‌کنند مورد نیاز می‌باشد.

الگوریتم پیشنهادی در این مقاله برای رسیدگی به مسائل مطرح شده، در یک محیط محاسبات ابری توسعه داده می‌شود. به‌منظور بهبود زمان تکمیل کلی، میانگین بهره‌وری ماشین مجازی، کاهش هزینه کل و نقض مهلت زمانی هر دو جنبه زمان‌بندی به‌صورت پویا و چندهدفه در نظر گرفته شده است. همچنین در نظر گرفتن نیازمندی‌های کیفی مختلف برای کاربران متفاوت مسئله مهمی است که مورد بررسی قرار گرفته شده است. همچنین خاصیت کشسانی ابر با توجه به نیاز به اجاره و آزادسازی ماشین مجازی جدید در زمان اجرا با دیگر خصوصیات مطرح شده نیز ترکیب شده است (بخش ۳).

۳ -روش پیشنهادی

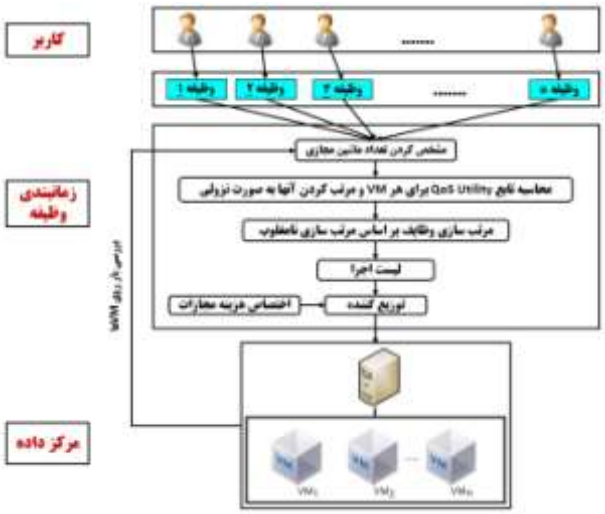
در محاسبات ابری، یک استخر بزرگ از منابع وجود دارد که شامل ماشین‌های مجازی فراوان است. استراتژی زمان‌بندی، توزیع وظایف محاسباتی به استخر منابع مجازی برای به دست آوردن قدرت محاسباتی، ذخیره‌سازی و انواع خدمات نرم‌افزار با توجه به نیازهای کاربر می‌باشد [۲۰].

در این مقاله یک مدل چندهدفه و پویا برای زمان‌بندی وظایف به‌منظور بهبود خصوصیات کیفی به‌صورت نشان داده شده در شکل (۱) ارائه می‌شود. توضیح مراحل اجرای زمان‌بندی در این مدل با توجه به الگوریتم ۱ ارائه می‌شود. به‌طورکلی مجموعه‌ای از انواع مختلف وظیفه با نرخ ورود متغیر، پارامترهای مهلت زمانی و هزینه اجرای وظیفه به‌عنوان ورودی‌های روش پیشنهادی در نظر گرفته می‌شوند و

www.SID.ir

Archive of SID

تعداد ماشین‌های مجازی، مجموعه‌ای از نتایج زمان‌بندی و تخصیص برای وظایف، خروجی‌ها هستند.



شکل ۱: استراتژی چندهدفه برای زمان‌بندی پویای وظایف با استفاده از منابع ابر کشسان

الگوریتم ۱. روش پیشنهادی

Input: A set of different types of tasks with variable entry rates, deadline and cost of tasks parameters
Output: A set of scheduling and allocation results for tasks.
1. Determine the number of virtual machines according to the tasks load that calculated by formula 1.
2. Calculate of the QoS_Utility(s) function for virtual machines by formula 3, and sort VMs descending order.
3. Sort tasks by non-dominated sorting based on deadline and task execution cost, that calculated by formula 4 and 5 (task execution cost calculated by formula 6).
4. **for** i ← 1 to Size of VM's list **do**
5. **for** j ← 1 to Size of task's list **do**
6. Calculate Normal_Rate by formula 9.
7. **if** VM_i_Normal_Rate < Threshold_rate **then**
8. Resource allocation $task_j$ to VM_i.
9. **end if**
10. **end for**
11. **end for**
12. a new set of task arrived
13. Calculate System_Total_Rate by formula 10.
14. **if** System_Total_Rate < Threshold_rate **then**
15. New arrived tasks are executable on available VMs.
16. **else** system is overloaded.
rent a virtual machine according to the calculated number by formula 12.
17. **end if**
18. **if** deadline missed **then**
19. Penalty cost= deadline missed seconds * penalty cost per second. (calculated by formula 13).
20. **end if**
21. **if** new VM load == 0 **then**
22. Delete VM.
23. **end if**
24. **return** a set of task scheduling results, VM counts.

گام ۱- در ابتدا بر اساس بار وظایف رسیده و با توجه به ورود درخواست‌ها با نرخ متغیر و عدم وجود ماشین مجازی، تعداد ماشین‌های مجازی با استفاده از فرمول (۱) مشخص می‌شود. بدین‌صورت که کل بار وظایف به دست آمده با توجه به فرمول (۲) محاسبه شده و بر بیشترین مقدار MIPS ماشین مجازی تقسیم می‌شود (خط ۱).

گام ۲- برای هر یک از ماشین‌های مجازی تابع QoS_Utility(s) توسط فرمول (۳) محاسبه شده و سپس بر اساس مقادیر به دست آمده، ماشین‌های مجازی به‌صورت نزولی مرتب می‌شوند (خط ۲).

گام ۳- مجموعه وظایف رسیده با روش مرتب‌سازی نامغلوب به‌صورت نزولی مرتب می‌شوند. این مرتب‌سازی بر اساس پارامترهای مهلت زمانی و هزینه اجرای وظیفه به‌صورت نشان داده شده در فرمول‌های (۴) و (۵) انجام می‌شود. هزینه اجرای وظیفه نیز با توجه به فرمول (۶) محاسبه می‌شود. به این صورت که هزینه هر ماشین (فرمول (۷)) در زمان اجرای وظیفه (فرمول (۸)) ضرب شده تا هزینه اجرای هر وظیفه به دست آید (خط ۳).

گام ۴- پس از مرتب‌سازی وظایف و ماشین‌های مجازی، وظایف با توجه به اولویت‌شان در لیست اجرا قرار می‌گیرند. اولین وظیفه از لیست اجرا به اولین ماشین مجازی اختصاص داده می‌شود. در این میان نرخ نرمال هر ماشین مجازی توسط فرمول (۹) محاسبه شده و با نرخ آستانه مقایسه می‌گردد. تا زمانی که نرخ نرمال بزرگ‌تر از نرخ آستانه نشود، به هر ماشین مجازی وظایف جدید تخصیص داده می‌شود. درواقع اگر بارکاری فعلی به‌علاوه بار کاری وظیفه بعدی از حد آستانه بیشتر شود، وظیفه جدید بر روی ماشین مجازی بعدی می‌رود. و این فرایند به همین ترتیب ادامه پیدا می‌کند تا کلیه وظایف به ماشین‌های مجازی تخصیص داده شوند (خط ۴-۱۱).

گام ۵- با رسیدن مجموعه وظایف بعدی، وظایف لیست اجرا مجدداً با در نظر گرفتن وظایف جدید مرتب می‌شوند. با توجه به فرمول (۱۰) نرخ کل سیستم محاسبه می‌گردد. سپس نرخ کل با نرخ آستانه مقایسه شده و اگر از نرخ آستانه کمتر شود یعنی وظایف جدید قابل اجرا روی ماشین‌های مجازی که در ابتدا ایجاد شده‌اند می‌باشند. در غیر این صورت سیستم دچار اضافه بار شده و نیاز به اجاره ماشین‌های مجازی جدید می‌باشد. قابل ذکر است که بیشترین بار کاری ماشین‌های موجود که در فرمول (۱۰) مورد استفاده قرار گرفته به‌وسیله فرمول (۱۱) محاسبه می‌گردد. همچنین تعداد ماشین‌های مجازی موردنیاز برای اجاره با توجه به فرمول (۱۲) محاسبه می‌شود. پس از آن ماشین‌های مجازی جدید نیز مرتب شده و به‌صورت شرح داده شده در قبل، وظایف به ماشین‌های مجازی اختصاص داده می‌شوند (خط ۱۲-۱۷).

گام ۶- در صورت نقض مهلت زمانی، به ازای هر تأخیر در زمان پاسخ به مشتری در حال سرویس، به ارائه‌دهنده یک جریمه مشخص بر اساس فرمول (۱۳) اختصاص داده می‌شود (خط ۱۸-۲۰).

گام ۷- در هر مرحله اگر بار ماشین‌های اجاره‌ای به صفر برسد یعنی ماشین‌های مجازی اولیه قادر به اجرای وظایف باشند، ماشین‌های اجاره‌ای، مازاد محسوب شده و حذف می‌شوند (خط ۲۱-۲۳).

www.SID.ir

Archive of SID

۳ ـ۱ ـتعیین تعداد ماشین‌های مجازی

برای استفاده به‌اندازه کافی از منابع در ابر و جلوگیری از عدم تحمیل هزینه اضافی، محاسبه مقدار واقعی منابع مورد نیاز امری ضروری می‌باشد. در این مدل، تعداد ماشین‌های مجازی با توجه به بار وظایف رسیده مشخص می‌شود. تعداد اولیه ماشین‌های مجازی موردنیاز از تقسیم جمع بار وظایف رسیده بر حداکثر مقدار MIPS ماشین‌های مجازی با توجه به فرمول (۱) محاسبه می‌گردد[۲۰].

$$Vm_Counts = \frac{Total_Task_Load}{Maximum_VM_Mips} \quad (۱)$$

برای به دست آوردن بار کل وظایف رسیده از جمع بار آن‌ها توسط فرمول (۲) استفاده می‌شود. قابل ذکر است بار وظایف با استفاده از مجموعه داده‌های PlanetLab در شبیه‌ساز ابر (CloudSim) مشخص شده است[۲۰].

$$Total_Task_Load = \sum_{k=1}^{n} Task_Load \quad (۲)$$

۳ ـ۲ ـمرتب‌سازی ماشین‌های مجازی

پس از تعیین تعداد مورد نیاز از ماشین‌های مجازی نیاز به مرتب‌سازی ماشین‌ها می‌باشد. QoS[۱۱] یا به عبارتی کیفیت سرویس برای شرح نیازمندی‌های کاربران مورد استفاده قرار می‌گیرد. کاربران مختلف به یک تنوعی از سرویس‌های محاسبات ابری نیاز دارند. به‌صورت یک شاخص کارایی خیلی مهم در محیط محاسبات ابری، پارامترهای کیفیت سرویس روی سرویس‌های ابر شامل چندین جنبه هستند و مطالعه حاضر زمان اجرای سرویس و هزینه اجرای سرویس را برای شرح QoS روی سرویس‌های منابع استفاده می‌کند.

تابع QoS_Utility(s) [۲۱] که از فرمول (۳) به دست آمده است برای نگاشت بردار خصوصیات QoS، $Qs= \{q_1(s), q_2(s), \ldots, q_r(s)\}$ روی هر سرویس کاندید به یک مقدار واقعی استفاده می‌شود.

$$QoS_Utility(s) = \sum_{k=1}^{r} \frac{Q_{j,k}^{max} - q_k(s)}{Q_{j,k}^{max} - Q_{j,k}^{min}} \times P_k \quad (۳)$$

درحالی که r تعداد خصوصیات کیفی ماشین مجازی یا سرویس می‌باشد. $q_k(s)$، k اُمین مقدار خصوصیت کیفی روی سرویس s می‌باشد. P_k اولویت کاربر است. $Q_{j,k}^{max}$ و $Q_{j,k}^{min}$ به ترتیب مقدار حداکثر و حداقل از k اُمین صفات QoS از تمام ماشین‌های مجازی کاندید است. پس از محاسبه تابع QoS_Utility(s) برای هر ماشین، با توجه به مقادیر به دست آمده برای هر ماشین مجازی، آن‌ها به‌صورت نزولی مرتب می‌شوند. به این دلیل که طبق روش پیشنهاد شده وظایف با هزینه و اندازه بیشتر به ماشین‌هایی با قدرت بیشتر ارسال می‌شوند.

۳ ـ۳ ـمرتب‌سازی وظایف با روش مرتب‌سازی نامغلوب

مرتب‌سازی نامغلوب برای حل مسائل چندهدفه استفاده شده است که در آن مسائل توابع به‌صورت چندهدفه با چندین معیار در یک زمان در نظر گرفته می‌شوند [۲۲]. زمانی که بحث از یک الگوریتم تک هدفه مطرح است، معیار برتری جواب‌ها نسبت به هم بسیار ساده و بدیهی است. زیرا تنها یک تابع هدف مدنظر می‌باشد و درصورتی‌که مسئله مورد بحث یک مسئله کمینه‌سازی باشد، جوابی که کمترین مقدار تابع هدف را دارا باشد مطلوب است و بر سایر جواب‌ها برتری دارد. اما زمانی که برای حل مسئله‌ای از یک الگوریتم چندهدفه استفاده می‌شود، به این معناست که حداقل دو تابع هدف مدنظر می‌باشد. به همین دلیل دیگر به‌آسانی نمی‌توان در مورد بعضی از جواب‌ها نظر قطعی داد. در اکثر موارد، نقاطی یافت می‌شود که هیچ‌کدام بر دیگری برتری کامل ندارد و نمی‌توان با مفهوم غلبه، دوبه‌دو بین آن‌ها مقایسه‌ای انجام داد. لذا برای به دست آوردن بهترین جواب‌ها باید آن‌ها را بر اساس یک معیاری مرتب کرد. در این روش به هر جواب یک رتبه[۱۲] اختصاص داده می‌شود که بر اساس تعداد مغلوب شدن آن‌ها نسبت به سایر نقاط انجام می‌شود. در انتها، نقاطی که بهترین رتبه یعنی رتبه ۱ و کمترین غلبه را دارا باشند به‌عنوان مجموعه جواب یا نقاط پارتو فرانت[۱۳] انتخاب می‌شوند.

هدف اصلی انتخاب یک وظیفه با حداقل اندازه و حداقل هزینه اجرا می‌باشد. دو تابع هدف به‌صورت نشان داده شده در فرمول (۴)و (۵) می‌باشند.

$$\begin{aligned} &Min\ f(Task_Size_k) \\ &= Task_Size_k | \forall j \exists i, f(Task_Size_i) \\ &\leq f(Task_Size_j) \end{aligned} \quad (۴)$$

$$\begin{aligned} &Min\ f(Task_execution_cost_k) \\ &= Task_execution_cost_k | \forall j \exists i, f(Task_execution_ \\ &\leq f(Task_execution_cost_j) \end{aligned} \quad (۵)$$

به‌طوری‌که

$$Task_Size, Task_execution_cost \in T\ (ID, Task_execution_cost, Task_Size)$$

و

$$i = \{1, 2, 3 \ldots . n\}, j = \{1, 2, 3 \ldots . n\}, k = \{1, 2, 3 \ldots . n\}$$

که در آن، Task_Size نشان‌دهنده اندازه وظیفه، Task_execution_cost نشان‌دهنده هزینه اجرای وظیفه، T مجموعه‌ای از وظایف به‌صورت $T=\{t_1, t_2, \ldots, t_n\}$ و n تعداد وظیفه است. مرتب‌سازی نامغلوب برای پیاده‌سازی الگوریتم زمان‌بندی وظیفه چندهدفه با هدف فوق استفاده می‌شود. در مرتب‌سازی نامغلوب، اهداف چندگانه در یک زمان عملی می‌شوند.

هزینه اجرای وظایف که یکی از اهداف در نظر گرفته شده در مرتب‌سازی نامغلوب است، به کمک فرمول (۶) محاسبه می‌گردد [۲۳].

www.SID.ir

Archive of SID

$$Task_execution_cost = \sum_{k \in SPp} Cost_of_VM_k \times Task_Execution_Time_k \quad (۶)$$

درحالی‌که SPp مجموعه ماشین‌های متعلق به p اُمین ارائه‌دهنده (SPp={k|VMk∈PthCloudprovider, P∈{1,2,…,cp}}) و cp تعداد ارائه‌دهندگان ابر است. $Cost\ of\ VM_k$، هزینه یک واحد CPU برای k اُمین ارائه‌دهنده است که از فرمول (۷) به دست می‌آید[۲۳]. و $Task_Execution_Time_k$، زمان اجرای وظایف اختصاص داده شده به VM_k است که از فرمول (۸) قابل محاسبه می‌باشد[۲۳].

$$Cost_of_VM = Cost_Per_Second() / Mips_Of_One_Pe() \quad (۷)$$

$$Task_Execution_Time_k = \sum_{i=1}^{n} \frac{Length \times PE + OutPutSize}{MIPS \times PE} \quad (۸)$$

۳ ۴ اختصاص منبع

در محاسبات ابری، مدیریت منابع کاملاً مبتنی بر تخصیص منابع است [۲۴]. پس از مرتب‌سازی وظایف و ماشین‌های مجازی، یکسری از وظایف با توجه به اولویتشان در لیست اجرا قرار می‌گیرند. سپس، اولین وظیفه از لیست اجرا به اولین ماشین مجازی در ابتدای لیست مرتب شده ماشین‌های مجازی اختصاص داده می‌شود. نرخ نرمال هر ماشین مجازی با توجه به فرمول (۹) محاسبه و با نرخ آستانه مقایسه می‌شود[۲۴]. تا زمانی که نرخ نرمال بزرگ‌تر از نرخ آستانه نشود، به هر ماشین مجازی وظایف جدید تخصیص داده می‌شود. درواقع اگر بارکاری فعلی به‌علاوه بار کاری وظیفه بعدی از حد آستانه بیشتر شود، وظیفه جدید بر روی ماشین مجازی بعدی می‌رود و این فرایند به همین ترتیب ادامه پیدا می‌کند تا کلیه وظایف به ماشین‌های مجازی تخصیص داده شوند[۲۴].

$$VM_Normal_Rate = \frac{Current_Workload_of_a_VM}{Maximum_VM_Mips} \quad (۹)$$

۳ ۵ اجاره و آزادسازی منابع

خاصیت کشسانی در محاسبات ابری "درجه‌ای است که یک سیستم قادر به انطباق با تغییرات حجم کار با ارائه و تخریب منابع به شیوه خودکار است، به‌طوری‌که در هر نقطه از زمان منابع موجود تا آنجا که ممکن است مطابق با تقاضای فعلی باشند" [۲۵]. در محیط ابر این خصوصیت برای جلوگیری از عدم تحمیل هزینه اضافی در خصوص نگهداری ماشین‌هایی که موردنیاز نمی‌باشند ارائه می‌گردد.

پس از اختصاص یکسری وظایف و با اضافه شدن وظایف جدید و افزایش حجم کار در زمان اجرا می‌توان هر زمان به ماشین‌های مجازی اضافه‌تر نیاز باشد، اقدام به اجاره تعدادی ماشین مجازی کرد و سپس، آن‌ها را جهت کاهش بار شبکه و جلوگیری از اتلاف منابع، آزاد کرد. بدین منظور پس از محاسبه بار کل وظایف، نرخ کل سیستم و حداکثر بار ماشین‌های موجود طبق فرمول‌های (۱۰) و (۱۱) محاسبه می‌شوند و می‌توان تعداد مورد نیاز از ماشین‌های مجازی را به دست آورد[۲۵].

$$System_Total_Rate = \frac{Total_Load}{Maximum_Workload_of_existed_VMs} \quad (۱۰)$$

$$Maximum_Workload_of_existed_VMs = \sum_{k=1}^{n} Maximum_VM_Mips \quad (۱۱)$$

نرخ کل سیستم برای محاسبه و اطلاع از اضافه بار محاسبه می‌شود. به این صورت که اگر نرخ کل بزرگ‌تر از نرخ آستانه (۰,۷) باشد ماشین‌های مجازی دچار اضافه بار شده و به تعداد محاسبه شده توسط فرمول (۱۲) ماشین مجازی اجاره می‌شود[۲۵].

$$Rental_Vm_Counts = \frac{System_Total_Rate - Maximum_Workload_of_existe}{Maximum_VM_Mips} \quad (۱۲)$$

قابل ذکر است، در صورت صفر شدن بار روی ماشین‌های اجاره‌ای، آن‌ها حذف و آزاد می‌شوند. اما با رسیدن وظایف سری بعد و نیاز مجدد به آن‌ها دوباره مورد استفاده قرار خواهند گرفت.

۳ ۶ اختصاص جریمه در صورت نقض مهلت زمانی

بسیاری از الگوریتم‌ها ممکن است هزینه را کاهش دهند ولی آن‌قدر مهلت زمانی را نقض کنند که این کاهش هزینه به چشم نیاید. بنابراین برای محاسبه هزینه در صورت نقض مهلت زمانی با تأثیر دادن جریمه در هزینه اجاره ماشین‌های مجازی، دیگر با اطمینان می‌توان گفت که هزینه کمتر نشان از موفقیت دارد. نقض مهلت زمانی، زمانی صورت می‌گیرد که زمان اجرای یک وظیفه از مقدار مهلت زمانی در نظر گرفته شده برای آن بیشتر شود. مهلت زمانی با توجه به فرمول (۱۴) محاسبه می‌شود [۲۶]. همچنین نرخ جریمه، هزینه مالی می‌باشد که ارائه‌دهنده برای هر تأخیر در درخواست مشتری در حال سرویس و در واحد زمان متحمل می‌شود. این نرخ جریمه برابر ۳$ در هر ثانیه فرض می‌شود [۲۷].

$$Penalty_Cost = Deadline_Missed_Seconds \times Penalty_Cost_Per_Second \quad (۱۳)$$

$$Deadline = \frac{Completion_Time}{load} \times 100 + Arrival \quad (۱۴)$$

www.SID.ir

Archive of SID

جدول ۱: نوع ماشین‌های مجازی

نوع ماشین مجازی	قدرت پردازشی	تعداد هسته	حافظه	پهنای باند	سایز	هزینه در هر ثانیه
۱	۱۰۰۰	۱	GB۲	۱۰۰ Mbit/s	G ۱۶۰ Disk	$۰/۱۲
۲	۱۰۰۰	۲	GB۴	۱۰۰ Mbit/s	G ۸۵۰ Disk	$۰/۴۸
۳	۱۰۰۰	۴	GB۸	۱۰۰۰ Mbit/s	G ۱۶۹۰ Disk	$۰/۹۶

۴ -ارزیابی

محاسبات ابری یک مفهوم متن‌باز[۱۴] برای شبیه‌سازی الگوریتم‌های زمان‌بندی ارائه می‌کند. این شامل بسیاری از مراکز داده و همه انواع منابع با استفاده از مجازی‌سازی برای ارائه خدمات به مشتری می‌باشد.

جدول ۲: نوع وظایف

نوع وظیفه	طول	سایز فایل	سایز خروجی
۱	۵۰۰۰۰	۱۰۰۰	۱۰۰۰
۲	۲۰۰۰۰۰	۳۰۰	۳۰۰
۳	۱۰۰۰۰۰	۱۰۰۰	۱۰۰۰

در این بخش به ارزیابی بهبود کیفیت سرویس با نتایج شبیه‌سازی شده می‌پردازیم. پارامترهای اصلی مورد ارزیابی عبارت‌اند از بهبود زمان تکمیل کلی، میانگین بهره‌وری ماشین مجازی، هزینه کل و نقض مهلت زمانی.

۴ ۱ -تنظیمات اولیه آزمایش‌ها

در این مقاله شبیه‌سازی با استفاده از ابزار CloudSim 3.0.3 [۲۸] برای تحقق بخشیدن به روش پیشنهادی انجام شد. در محیط ابر هر وظیفه معادل یک cloudlet در محیط شبیه‌ساز می‌باشد. نرخ ورود وظایف بر اساس توزیع پوآسون با نرخ λ=۳ است. این آزمایش با نرخ لانداهای مختلف آزمایش شد ولی به خاطر محدود بودن فضا به‌عنوان نمونه فقط نتایج λ=۳ در نتایج نشان داده شد. ماشین‌های مجازی و وظایف به سه گروه بر طبق جداول (۱) و (۲) تقسیم می‌شوند که این دسته‌بندی از مرجع [۲۷] الهام گرفته شده است. به‌منظور شبیه‌سازی تعداد ۱۰۰، ۲۰۰، ۴۰۰، ۶۰۰، ۸۰۰ و ۱۰۰۰ وظیفه با الهام از [۲۹] به‌عنوان درخواست‌های ورودی در نظر گرفته می‌شوند. ماشین‌های مجازی براساس زمان و هزینه اجرای سرویس مرتب‌سازی می‌شوند. وظایف بر اساس دو پارامتر مهلت زمانی و هزینه به روش نامغلوب مرتب شده؛ همچنین تعداد ماشین‌های مجازی به‌صورت پویا و براساس بار وظایف تعیین می‌شوند. وظایف به هر ماشین با نرخ آستانه (۰/۹) اختصاص داده می‌شوند. پس از اختصاص وظایف به ماشین‌های مجازی و با ورود وظایف جدید، اگر بار کل روی ماشین‌ها بیشتر از مقدار آستانه (۰/۷) شود [۳]، سیستم نیاز به ماشین مجازی مازاد پیدا می‌کند. با توجه به خصوصیت کشسانی ابر، استراتژی زمان‌بند پیشنهادی قادر به اجاره منابع مازاد و سپس آزادسازی منابع مورد نیاز می‌باشد. درنهایت در صورت وجود تأخیر در پاسخ به درخواست مشتری و نقض مهلت زمانی به ازای هر واحد زمان، فراهم‌کننده متحمل جریمه خواهد شد.

۴ ۲ -معیارها برای ارزیابی الگوریتم پیشنهاد شده

معیارهای استفاده شده برای بررسی استراتژی زمان‌بندی پیشنهادی زمان تکمیل کلی، بهره‌وری ماشین مجازی، هزینه کل و نقض مهلت زمانی است که در ادامه شرح داده می‌شوند.

- زمان تکمیل کلی: زمان تکمیل اجرای یک وظیفه است. این شامل زمان اجرا و تأخیر ناشی از پاسخگویی سیستم ابر است [۳۰].
- هزینه کل: هزینه کل با توجه به جمع هزینه هر ماشین مجازی در زمانی که ماشین مجازی مشغول به کار بوده محاسبه می‌شود. به دلیل اینکه در استراتژی پیشنهادی هزینه جریمه نیز در نظر گرفته شده است، در پارامتر هزینه کل علاوه بر در نظر گرفتن هزینه ماشین مجازی، هزینه تحمیل شده در اثر جریمه نقض مهلت زمانی نیز لحاظ خواهد شد.
- بهره‌وری ماشین مجازی: ظرفیتی از منابع که در طول اجرا در حال استفاده بوده و تحت عنوان بهره‌وری منابع شناخته می‌شود. این معیار با توجه به فرمول (۱۵) محاسبه شده است [۲۵]. بار هر ماشین در بازه زمانی که در حال اجرا بوده تقسیم بر تعداد کل ماشین‌ها میانگین بهره‌وری ماشین مجازی را می‌دهد. درحالی‌که $L_{i,j}$ بار روی ماشین مجازی است و m تعداد کل ماشین‌ها شامل تعداد ماشین‌های اولیه به‌علاوه ماشین‌های اجاره‌ای می‌باشد[۲۷].

$$VM_Utilization_Mean = \sum_{i=1}^{m} \frac{\int_{s_i}^{f_i} L_{i,j}}{(Finish_Time_of_each_VM_i - Start_Time_of_each_VM_i)} \Big/ m \quad (۱۵)$$

- نقض مهلت زمانی: اگر زمان پایان هرکدام از وظایف بزرگ‌تر از مهلت زمانی تعیین شده کاربر باشد موجب نقض مهلت زمانی می‌شود. این پارامتر در این تحقیق رابطه مستقیم با هزینه کل دارد. این پارامتر بدین منظور تعریف می‌شود که بتوان با اطمینان گفت هزینه کمتر نشان از موفقیت دارد. بدین‌صورت که بسیاری از الگوریتم‌ها ممکن است هزینه را کاهش دهند ولی آن‌قدر مهلت زمانی را نقض کنند که این کاهش هزینه به چشم نیاید. به همین دلیل به ازای هر ثانیه نقض مهلت زمانی سیستم متحمل جریمه می‌شود. بدیهی است، یک تأخیر کوچک ممکن است باعث نقض یک مهلت زمانی شود.

۴ ۳ -ارزیابی آزمایش‌ها

برای ارزیابی عملکرد روش پیشنهادی، ۴ معیار مهم برای ارزیابی در نظر گرفته شده است که عبارت‌اند از: زمان تکمیل کلی، میانگین

www.SID.ir

Archive of SID

بهره‌وری ماشین مجازی، هزینه کل و نقض مهلت زمانی. علاوه بر این، استراتژی پیشنهادی در آزمایش‌ها با الگوریتم‌های MultiObjective [۱۲]، FCFS، Min-Min بهبود یافته [۳۱]، Priority Scheduling [۳۲] و MOF [۳۳] مقایسه و ارزیابی می‌شود. الگوریتم FCFS[۱۵] به هر فرایند، زمانی از زمان پردازنده را مطابق با ترتیب ورودش اختصاص می‌دهد. در الگوریتم Min-Min بهبود یافته وظیفه با کمترین زمان تکمیل، انتخاب شده و به ماشین متناظر اختصاص داده می‌شود. الگوریتم Priority Scheduling برای هر فرآیند یک اولویت تعیین می‌کند و فرآیند با اولویت بالا مجاز به اجرا می‌باشد. الگوریتم MultiObjective وظایف را با مرتب‌سازی نامغلوب مرتب کرده و سپس به‌صورت پی‌درپی به ماشین‌هایی که بر اساس MIPS به‌صورت نزولی مرتب شده اختصاص می‌دهد. وجه تشابه این روش با روش پیشنهادی در استفاده از مرتب‌سازی نامغلوب می‌باشد. اما وجه تمایز آن عبارت است از تعداد ماشین‌های مجازی که در روش پیشنهادی به‌صورت پویا تعیین می‌شود. در روش پیشنهادی ماشین‌های مجازی براساس نیازمندی‌های کاربر مانند زمان و هزینه اجرای سرویس در نظر گرفته می‌شود. درحالی‌که این روش فقط ماشین‌ها را براساس MIPS آن‌ها به‌صورت نزولی مرتب می‌کند. در روش پیشنهادی برای نقض مهلت زمانی جریمه در نظر گرفته شده و درنهایت، روش پیشنهادی از خاصیت کشسانی ابر برای آزادسازی منابع استفاده می‌کند که هیچ‌کدام از این موارد در روش MultiObjective مورد استفاده قرار نگرفته است. الگوریتم MOF معادل روش Objective Functions Multi در مرجع می‌باشد. روش MOF در ابتدا ماشین‌های مجازی را بر اساس پارامترهای کیفیت سرویس (MIPS و اندازه دانه[۱۶]) و همچنین وظایف را نیز با روش مرتب‌سازی نامغلوب مرتب کرده، سپس وظایف را به ماشین‌های مجازی اختصاص می‌دهد. به این صورت که تعداد وظایف را بر تعداد ماشین‌های مجازی تقسیم کرده و به همان تعداد وظایف را به ماشین‌های مجازی اختصاص می‌دهد.

با توجه به نتایج نشان داده شده در شکل (۲) استراتژی پیشنهادی با توجه به پارامتر زمان تکمیل کلی در مقایسه با الگوریتم‌های MultiObjective، FCFS، Min-Min بهبود یافته، Priority Scheduling و MOF ارزیابی می‌شود. همان‌طورکه در شکل (۲) قابل مشاهده می‌باشد با تعداد ۱۰۰ وظیفه روش پیشنهادی در مقایسه با دیگر الگوریتم‌ها بدتر عمل کرده و بهبودی حاصل نشده است. درواقع زمانی که آزمایش‌ها با تعداد کمی از وظایف انجام می‌شوند، به دلیل استفاده از مرتب‌سازی نامغلوب در روش پیشنهادی، خود عملیات مرتب‌سازی زمان‌بر و دارای سربار می‌باشد. اما زمانی که تعداد وظایف زیاد می‌شود این سربار در مقایسه با حجم کار، ناچیز محسوب شده و باعث بهبود چشمگیر روش پیشنهادی در مقایسه با سایر روش‌ها می‌شود. به‌عنوان‌مثال با ۲۰۰ وظیفه، از نتایج می‌توان مشاهده کرد که روش پیشنهادی فقط در مقایسه با MultiObjective، FCFS، Priority Scheduling و MOF به ترتیب ۱۲/۵۲، ۱۲/۵۶، ۲۴/۹۴ و ۶/۱۷ ٪ بهبود داشته و با ۴۰۰ وظیفه در مقایسه با MultiObjective، FCFS، Min-Min بهبود یافته، Priority Scheduling و MOF به ترتیب ۴۶/۷۴، ۵۵/۴۱، ۱۹/۶۱، ۵۳/۴۳ و ۳۹/۲ ٪ زمان تکمیل کمتری ارائه می‌دهد. می‌توان گفت زمان تکمیل کلی با نرخ درخواست و تعداد مناسب ماشین مجازی در بین فعالیت‌های فرآیند می‌تواند تحت تأثیر قرار گیرد.

در روش پیشنهادی تعداد ماشین‌های مجازی با توجه به بار وظایف تعیین می‌شود. بنابراین در بیشتر موارد تعداد وظایف متناسب با ماشین‌های مجازی است که می‌تواند موجب بهبود زمان تکمیل کلی شود. همچنین به دلیل اینکه در روش پیشنهادی وظایف با هزینه و اندازه بیشتر بر روی ماشین‌هایی با قدرت بیشتر ارسال می‌شوند بنابراین می‌تواند عملکرد بهتری نسبت به سایر الگوریتم‌ها در این پارامتر از خود نشان دهد.

پارامتر میانگین بهره‌وری ماشین مجازی که در شکل (۳) با پنج الگوریتم دیگر مورد مقایسه قرار گرفته است در مقایسه با الگوریتم Min-Min بهبود یافته فقط در آزمایش با ۱۰۰ وظیفه بهبود ۳/۶۱ درصدی داشته و در بقیه موارد الگوریتم Min-Min بهبود یافته بهتر است. ولی با این وجود روش پیشنهادی در تمام موارد از الگوریتم‌های MultiObjective، FCFS، Priority Scheduling و MOF بهره‌وری بهتری داشته است. یکی از دلایل بهبود این پارامتر در روش پیشنهادی به دلیل تأمین منابع مبتنی بر تقاضا، یعنی اجاره و آزادسازی منابع موردنیاز با توجه به خصوصیت کشسانی ابر است. بدین‌صورت که ماشین‌های مازاد اجاره شده هنگامی‌که بار روی آن‌ها به صفر برسد آزاد شده و وجود ماشین‌های اضافه باعث پایین آمدن بهره‌وری کل نمی‌گردد. قابل ذکر است، علت نوسان به وجود آمده در نتایج الگوریتم پیشنهادی طرز چینش وظایف در ماشین‌های مجازی می‌باشد.

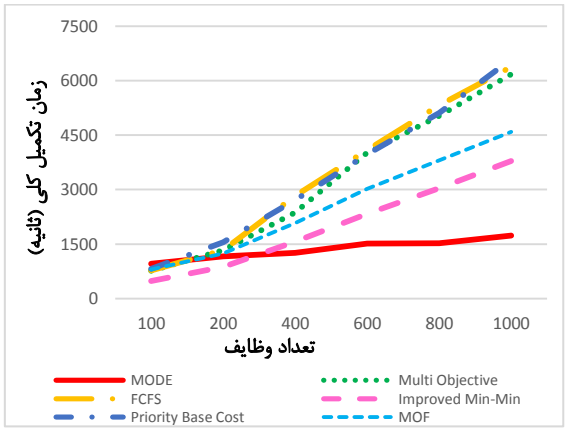
از مشاهده شکل (۴) در رابطه با پارامتر نقض مهلت زمانی به این نتیجه می‌توان رسید که استراتژی پیشنهادی در مقایسه با MultiObjective، FCFS، Min-Min بهبود یافته، Priority Scheduling و MOF نقض مهلت زمانی کمتری دارد. یکی از دلایل ارزیابی این پارامتر به دلیل اطمینان از کاهش هزینه است. به همین دلیل ارتباط مستقیم با هزینه دارد. به این صورت که در محاسبه هزینه در ازای هر ثانیه نقض مهلت زمانی برنامه متحمل هزینه جریمه می‌شود. چون بسیاری از الگوریتم‌ها ممکن است هزینه را کاهش دهند ولی آن‌قدر مهلت زمانی را نقض کنند که این کاهش هزینه به چشم نیاید. نکته قابل‌توجه در ارزیابی روش پیشنهادی در شکل (۴) این است که مقدار نقض مهلت زمانی با تعداد وظایف مختلف تقریباً در یک محدوده بوده است. می‌توان گفت به این دلیل است که الگوریتم پیشنهادی متناسب با هر تعداد وظایف رسیده ماشین مجازی مناسب اجاره و یا آزاد می‌کند.

با نتایج به دست آمده از شکل (۵) می‌توان بهبود قابل‌توجهی را در استراتژی پیشنهادی با توجه به پارامتر هزینه کل مشاهده کرد.

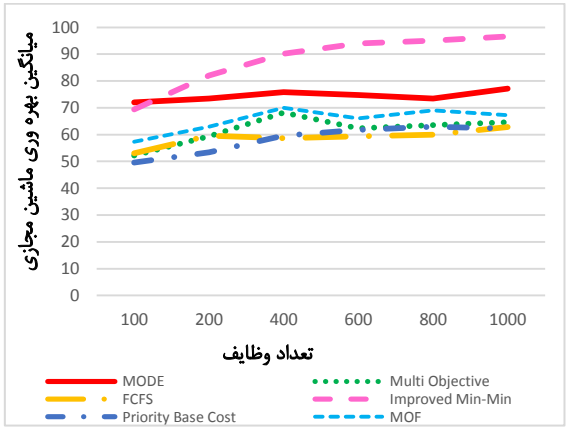
www.SID.ir

Archive of SID

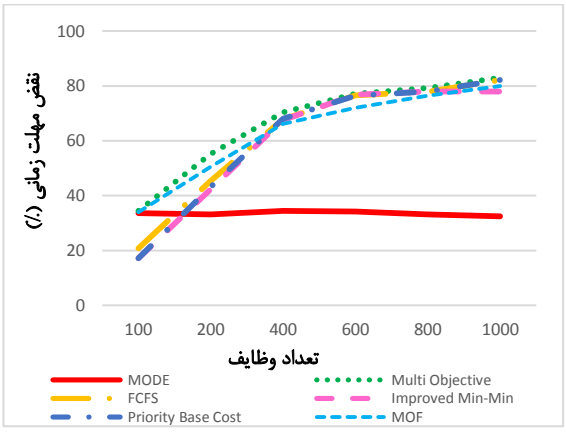
یکی از دلایل اولیه این بهبود این می‌تواند باشد که الگوریتم پیشنهادی در مراحل مختلف پارامتر هزینه را در نظر گرفته است. بدین‌صورت که این استراتژی الگوریتمی چندهدفه بوده و یکی از معیارهای که در نظر گرفته است، معیار هزینه می‌باشد. همچنین در انتخاب ماشین مجازی یکی از معیارهای انتخاب ماشین مجازی نیز هزینه می‌باشد. از طرف دیگر به دلیل اینکه به ازای هر ثانیه نقض مهلت زمانی، برنامه متحمل هزینه جریمه می‌شود و با توجه به نتایج شکل (۵) که الگوریتم پیشنهادی کمترین مقدار نقض مهلت زمانی را دارد، نسبت به دیگر الگوریتم‌ها کمترین مقدار هزینه را نیز داراست. به همین دلیل الگوریتم MultiObjective نسبت به دیگر الگوریتم‌ها هزینه بیشتری متحمل شده است.



شکل ۲: نتایج ارزیابی پارامتر زمان تکمیل کلی (تعداد ماشین مجازی متغیر(۲۳-۱۷۸) و نرخ ورود متغیر بر اساس توزیع پوآسون)

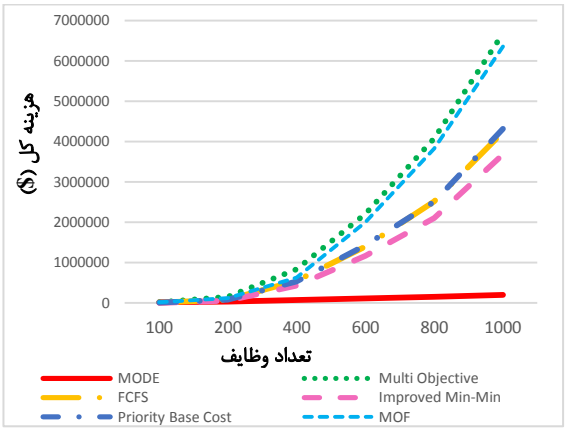


شکل ۳: نتایج ارزیابی پارامتر بهره‌وری ماشین مجازی (تعداد ماشین مجازی متغیر (۲۳-۱۷۸) و نرخ ورود متغیر بر اساس توزیع پوآسون)



شکل ۴: نتایج ارزیابی پارامتر نقض مهلت زمانی (تعداد ماشین مجازی متغیر(۲۳-۱۷۸) و نرخ ورود متغیر بر اساس توزیع پوآسون)

به‌طورکلی از آزمایش‌های انجام شده می‌توان به این نتیجه رسید که الگوریتم پیشنهادی در اکثر موارد موجب بهبود پارامترهای زمان تکمیل کلی، میانگین بهره‌وری ماشین مجازی، هزینه کل و نقض مهلت زمانی می‌شود.



شکل ۵: نتایج ارزیابی پارامتر هزینه کل (تعداد ماشین مجازی متغیر(۲۳-۱۷۸) و نرخ ورود متغیر بر اساس توزیع پوآسون)

۵ -نتیجه

مسئله زمان‌بندی وظیفه در رایانش ابری، سعی در ارائه یک زمان‌بندی بهینه برای اجرای وظایف و تخصیص بهینه منابع دارد. به علت روند رو به رشد محاسبات ابری، مطالعه هر چه بیشتر در زمینه زمان‌بندی وظیفه در محیط ابر امری مهم و ضروری است. یکی از جنبه‌های مهم محاسبات ابری زمان‌بندی پویای تعداد زیادی درخواست‌های وظیفه است که به‌وسیله کاربران ارائه می‌شوند و مراکز داده ابر نه‌تنها باید این وظیفه‌های عظیم را اجرا کنند بلکه باید نیازمندی‌های چندگانه کاربران مختلف را ارضاء کنند. اهداف چندگانه ممکن است ماهیت متضاد داشته باشند. برای نمونه، حداقل کردن زمان تکمیل کلی و نقض مهلت زمانی روی تعداد زیادی وظایف دشوار است درحالی‌که هزینه‌ها کاهش پیدا کند. بهینه‌سازی این اهداف به خاطر شرایط زمان اجرای نامشخص از قبیل خاصیت کشسانی روی منابع و توزیع‌های بار

www.SID.ir

Archive of SID

کاری متغیر دشوار است. در این مقاله، ابتدا مهم‌ترین الگوریتم‌های زمان‌بندی و روش‌های موجود مورد بررسی قرار گرفتند. سپس یک استراتژی زمان‌بندی پویای وظایف برای بهبود خصوصیات کیفی ارائه شد که از خصوصیت کشسانی منابع ابر در حین زمان‌بندی استفاده می‌کند. در این استراتژی، ابتدا با ورود درخواست‌ها با نرخ متغیر، تعداد ماشین‌های مجازی براساس بار وظایف رسیده مشخص می‌شوند. سپس ماشین‌های مجازی با در نظر گرفتن نیازمندی‌های کاربر و وظایف به‌صورت چندهدفه مرتب خواهند شد و وظایف با در نظر گرفتن ظرفیت ماشین‌های مجازی، به آن‌ها به‌طور پویا اختصاص داده شده و اجرا خواهند شد. لازم به ذکر است در حین اجرا، با توجه به بار وظایف رسیده و با استفاده از خصوصیت کشسانی ابر در صورت نیاز ماشین مجازی جدید اجاره و در صورت عدم نیاز، آزادسازی می‌شوند. در آخر در صورت وجود تأخیر در پاسخ به درخواست مشتری به ازای هر واحد زمان فراهم‌کننده متحمل جریمه خواهد شد. نتایج ارزیابی نشان می‌دهد که روش پیشنهادی زمان‌بندی بهره‌ورتری را در پارامترهای کیفیت سرویس از قبیل زمان تکمیل کلی، میانگین بهره‌وری ماشین مجازی، هزینه کل و نقض مهلت زمانی نسبت به الگوریتم‌های FCFS، Priority Scheduling، Min-Min بهبود یافته، MultiObjective و MOF ارائه می‌دهد. در آینده تلاش خواهیم کرد تا پارامترهای کیفیت سرویس بیشتری برای وظایف چندهدفه در مرتب‌سازی نامغلوب استفاده کنیم. نیازهای متفاوت‌تر و بیشتری از کاربران متفاوت را در مرتب‌سازی ماشین‌های مجازی در نظر خواهیم گرفت. همچنین در مرحله تخصیص منابع، می‌توان قبل از تخصیص وظایف، ماشین‌های مجازی را بررسی کرد به‌گونه‌ای که بیشترین حجم وظیفه ممکن را در ماشین مجازی اشغال کند به‌طوری‌که کمترین فضای بلااستفاده باقی بماند و سپس وظایف را تخصیص داد. در آینده می‌توان رویکردهای گزینش ماشین مجازی مازاد و یا ترکیبی از آن‌ها را مورد مطالعه قرار داد.

مراجع

[1] H. Jiao, J. Zhang, J. Li, J. Shi, and J. Li, "Immune optimization of task scheduling on multidimensional QoS constraints," *Cluster Comput.*, vol. 18, no. 2, pp. 909–918, 2015.

[2] S. K. Panda, I. Gupta, and P. K. Jana, "Task scheduling algorithms for multi-cloud systems: allocation-aware approach," *Inf. Syst. Front.*, pp. 1–19, 2017.

[3] L. Ma, Y. Lu, F. Zhang, and S. Sun, "Dynamic Task Scheduling in Cloud Computing Based on Greedy Strategy," *Proc. Int. Conf. Trust. Comput. Serv.*, pp. 156–162, 2012.

[4] L. Tang, J.-S. Pan, Y. Hu, P. Ren, and Y. T. and H. Zhao, "A Novel Load Balance Algorithm for Cloud Computing," © *Springer Int. Publ. Switz. 2016*, vol. 329, pp. 325–333, 2016.

[5] K. Dubey, M. Kumar, and S. C. Sharma, "Modified HEFT Algorithm for Task Scheduling in Cloud Environment," *Procedia Comput. Sci.*, vol. 125, pp. 725–732, 2018.

[6] R. Moreno-Vozmediano, R. S. Montero, E. Huedo, and I. M. Llorente, "Orchestrating the Deployment of High Availability Services on Multi-zone and Multi-cloud Scenarios," *J. Grid Comput.*, pp. 1–15, 2017.

[7] M. Ghetas and C. H. Yong, "Resource Management Framework for Multi-tier Service Using Case-Based Reasoning and Optimization Algorithm," *Arab. J. Sci. Eng.*, 2017.

[8] L. Chen, M. Qiu, J. Song, Z. Xiong, and H. Hassan, "E2FS: an elastic storage system for cloud computing," *J. Supercomput.*, pp. 1–16, 2016.

[9] R. Khorsand, F. Safi-Esfahani, N. Nematbakhsh, and M. Mohsenzade, "Taxonomy of Workflow Partitioning Problems and Methods in Distributed Environments," *J. Syst. Softw.*, 2017.

[10] K. R. R. Babu and P. Samuel, "Enhanced Bee Colony Algorithm for Efficient Load Balancing and Scheduling in Cloud," © *Springer Int. Publ. Switz. 2016*, vol. 237, pp. 67–78, 2016.

[11] H. Gamal El Din Hassan Ali, I. A. Saroit, and A. M. Kotb, "Grouped tasks scheduling algorithm based on QoS in cloud computing network," *Egypt. Informatics J.*, vol. 18, no. 1, pp. 11–19, 2017.

[12] A. V. Lakra and D. Kumar Yadav, "Multi-objective tasks scheduling algorithm for cloud computing throughput optimization," *Procedia Comput. Sci.*, vol. 48, no. C, pp. 107–113, 2015.

[13] M. S. Aslanpour, M. Ghobaei-Arani, and A. N. Toosi, "Auto-scaling Web Applications in Clouds: A Cost-Aware Approach," *J. Netw. Comput. Appl.*, p. , 2017.

[14] L. D. Dhinesh Babu and P. Venkata Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Appl. Soft Comput. J.*, vol. 13, no. 5, pp. 2292–2303, 2013.

[15] U. Bhoi and P. Ramanuj, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing," *Int. J. Appl. or Innov. ...*, vol. 2, no. 4, pp. 259–264, 2013.

[16] M. Beltr´an, "BECloud: A new approach to analyse elasticity enablers of cloud services," *Futur. Gener. Comput. Syst.*, vol. 64, pp. 39–49, 2016.

[17] S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *Futur. Gener. Comput. Syst.*, vol. 29, no. 4, pp. 1012–1023, 2013.

[18] M. Ghobaei-Arani, S. Jabbehdari, and M. A. Pourmina, "An autonomic approach for resource provisioning of cloud services," *Cluster Comput.*, vol. 19, no. 3, pp. 1017–1036, 2016.

[19] M. Ghobaei-Arani, S. Jabbehdari, and M. A. Pourmina, "An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach," *Futur. Gener. Comput. Syst.*, 2016.

[20] B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," *Adv. Eng. Softw.*, vol. 42, no. 7, pp. 419–425, 2011.

[21] H. Li, S. Ge, and L. Zhang, "A QoS-based scheduling algorithm for instance-intensive workflows in cloud environment," *26th Chinese Control Decis. Conf. CCDC 2014*, pp. 4094–4099, 2014.

[22] J. Zhao, W. Zeng, M. Liu, and G. Li, "Multi-objective Optimization Model of Virtual Resources Scheduling Under Cloud Computing and It's Solution," *Cloud Serv. Comput.*, pp. 185–190, 2011.

[23] F. Ramezani, J. Lu, J. Taheri, and F. K. Hussain, "Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments," *World Wide Web*, vol. 18, no. 6, pp.

www.SID.ir

Archive of SID

[29] R. Khorsand, F. Safi-Esfahani, N. Nematbakhsh, and M. Mohsenzade, "ATSDS: adaptive two-stage deadline-constrained workflow scheduling considering run-time circumstances in cloud computing environments," *J. Supercomput.*, vol. 73, no. 6, pp. 2430–2455, 2016.

[30] S. Banerjee, M. Adhikari, S. Kar, and U. Biswas, "Development and Analysis of a New Cloudlet Allocation Strategy for QoS Improvement in Cloud," *Arab. J. Sci. Eng.*, vol. 40, no. 5, pp. 1409–1425, 2015.

[31] G. Wang and H. C. Yu, "Task Scheduling Algorithm Based on Improved Min-Min Algorithm in Cloud Computing Environment," *Appl. Mech. Mater.*, vol. 303–306, pp. 2429–2432, 2013.

[32] M. Dakshayini and H. S. Guruprasad, "An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment," *Int. J. Comput. Appl.*, vol. 32, no. 9, pp. 975–8887, 2011.

[33] A. Narwal and S. Dhingra, "Task Scheduling Algorithm Using Multi-Objective Functions for Cloud Computing Environment," *2nd Int. Conf. Sustain. Comput. Tech. Eng.*, vol. 27–28, no. Vm, 2017.

1737–1757, 2015.

[24] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid, "Recent advancements in resource allocation techniques for cloud computing environment: a systematic review," *Cluster Comput.*, vol. 20, no. 3, pp. 2489–2533, 2017.

[25] N. R. Herbst, S. Kounev, and R. Reussner, "Elasticity in Cloud Computing : What It Is , and What It Is Not," *Present. as part 10th Int. Conf. Auton. Comput.*, pp. 23–27, 2013.

[26] F. Cottet, J. Delacroix, Z. Mammeri, and C. Kaiser, *Scheduling in real-time systems*. 2002.

[27] L. Wu, S. Kumar Garg, S. Versteeg, and R. Buyya, "SLA-based Resource Provisioning for Hosted Software as a Service Applications in Cloud Computing Environments," *IEEE Trans. Serv. Comput.*, vol. 7, no. 3, pp. 465–485, 2014.

[28] C. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, esar A. F. De Rose, and and R. Buyya, "CloudSim: a toolkit formodeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw. - Pract. Exp.*, vol. 41, no. 1, pp. 23–50, 2011.

زیرنویس‌ها

[1] Elasticity
[2] Honey Bee Behavior inspired Load Balancing
[3] Platform
[4] Analytical Hierarchical Process
[5] Monitor
[6] Analyze
[7] Plan
[8] Execute
[9] Autonomic
[10] Reinforcement Learning (RL)
[11] Quality of service
[12] Rank
[13] Parto frante
[14] Open source
[15] First Come First Served
[16] Granularity size

www.SID.ir